1-1-2006

Microcosm: a lost cost 3-D wireless sensor test-bed within a controllable environment

David Marsh
University College Dublin

Richard Tynan
University College Dublin

Stephen T. Beirne
University of Wollongong, sbeirne@uow.edu.au

Roderick L. Shepherd
University of Wollongong, rods@uow.edu.au

Gregory O'Hare
University College Dublin

See next page for additional authors

Recommended Citation

Marsh, David; Tynan, Richard; Beirne, Stephen T.; Shepherd, Roderick L.; O'Hare, Gregory; Diamond, Dermot; and Corcoran, Brian: Microcosm: a lost cost 3-D wireless sensor test-bed within a controllable environment 2006, 820-834.
https://ro.uow.edu.au/scipapers/4239

Microcosm: a lost cost 3-D wireless sensor test-bed within a controllable environment

Abstract

This paper describes the creation of Microcosm, a low cost wireless sensor network test-bed within a controlled environment to facilitate WSN experiments in three dimensions, with an emphasis on executing sensing-related experiments. The design of the sensing hardware, software, support tools and the experimental environment itself are given. Issues related to the design of this configuration are discussed, with the potential pitfalls and eventual solutions alike given. Finally, current and future uses for the test-bed are listed.

Keywords

test, lost, wireless, 3, cost, environment, within, bed, sensor, microcosm, controllable

Disciplines

Life Sciences | Physical Sciences and Mathematics | Social and Behavioral Sciences

Publication Details

Marsh, D., Tynan, R., Beirne, S. T., Shepherd, R. L., O'Hare, G., Diamond, D. & Corcoran, B. (2006). Microcosm: a lost cost 3-D wireless sensor test-bed within a controllable environment. In Mobile Ad-hoc and Sensor Networks: Proceedings, 13-15 December, Hong Kong. Lecture Notes in Computer Science, 4325 820-834.

Authors

David Marsh, Richard Tynan, Stephen T. Beirne, Roderick L. Shepherd, Gregory O'Hare, Dermot Diamond, and Brian Corcoran

Microcosm: A Low Cost 3-D Wireless Sensor Test-Bed Within a Controllable Environment

David Marsh[1], Richard Tynan[1], Stephen Beirne[2], Roderick Shepherd[3], Gregory O'Hare[1], Dermot Diamond[3], and Brian Corcoran[2]

[1] School of Computer Science and Informatics, UCD Dublin
Belfield, Dublin 4, Ireland
{david.marsh, richard.tynan, gregory.ohare}@ucd.ie
http://www.cs.ucd.ie/csprism/index.html
[2] School of Mechanical and Manufacturing Engineering,
Dublin City University
Glasnevin, Dublin 9, Ireland
{stephen.beirne3, brian.corcoran}@dcu.ie
[3] National Centre for Sensor Research,
Dublin City University
Glasnevin, Dublin 9, Ireland
{roderick.shepherd, dermot.diamond}@dcu.ie

Abstract. This paper describes the creation of Microcosm, a low cost wireless sensor network test-bed within a controlled environment to facilitate WSN experiments in three dimensions, with an emphasis on executing sensing-related experiments. The design of the sensing hardware, software, support tools and the experimental environment itself are given. Issues related to the design of this configuration are discussed, with the potential pitfalls and eventual solutions alike given. Finally, current and future uses for the test-bed are listed.

1 Introduction

Since the potential applications of wireless sensor networks are so diverse, a similar diversity is reflected in test-beds for WSN experimentation. Despite the potential differences, some lessons from the construction of any particular test-bed should be applicable to many other set ups. As yet, not all possible classes of WSN test-beds have been explored, and hence not all problems have been investigated. This paper describes the design of a novel test-bed designed to accommodate approximately 150 individual sensors in a controlled test chamber, along with the necessary support software.

As with many aspects of wireless sensors networks, the design of a test-bed for reliable, recordable and repeatable experimentation is fraught with both expected and unexpected problems, trade-offs and compromises. Additionally, one of the major drawbacks of implementing a large scale sensor network is the cost. Even relatively old technology, for instance the Mica2 mote [1], still costs over $100 for the processing/radio unit alone, with sensing modules priced higher again. Lower cost alternatives exist, but are often far bulkier due to being based on dual inline packaging components. SmartIts [2] are one such example and are generally easy to customise, and so represent a useful

J. Cao et al. (Eds.): MSN 2006, LNCS 4325, pp. 820–834, 2006.

alternate avenue for exploration. This paper describes the issues experienced during the construction a low cost test-bed, one which demonstrates that complex, high resolution sensory data can be collected and used in a feedback-based control mechanism, without requiring heavy financial investment. By detailing those experiences, it is hoped that others who may wish to undertake the fabrication of a WSN test-bed will be able to avoid some of the obstacles that were encountered.

Microcosm is tailored towards experiments in the sensing domain. Having a controlled environment rules out the use of open systems, typified by sensor node deployments throughout buildings or outdoors, and thus a carefully constructed test chamber was employed, one in which the conditions could be altered as required, hence the name Microcosm. Since most WSN sensing work has dealt with planar configurations of sensors, it was decided that a high-density 3-D arrangement of nodes would provide new avenues for experimentation. Networking is not the primary concern, and the design choices reflect this relationship. Methods for increasing sensor density without incurring significant additional costs are given. Additionally, a discussion of requirements for closing the control loop, one of the major focuses of research using WSNs, is given.

The remainder of this paper is organised as follows. The next section lists a set of assumptions and design constraints that influenced the design of Microcosm. Section 3 lists the desired features of the test-bed, and describes some of the obstacles that were encountered, and the design choices used to overcome them. Section 4 provides comparative information on other test-beds. Finally, a brief section about the future work that this system will support is given.

2 Design Constraints and Assumptions

In the construction of any sensor network, certain assumptions must be made about the operation of the network. For instance, the sensing tasks it will be required to perform, the specifications for the communications channels, and the method of access to the WSN all contribute to the system requirements at the design stage. For this test-bed, the following constraints governed the design process.

- *Priority is sensory experimentation:* While networking experiments are planned, the primary use for Microcosm is in the sensing domain. Thus resources should be focused on creating a network that can produce good quality sensor data of many different types. As opposed to many wireless sensor networks, where individual sensor nodes are separated by distances on the order of meters, a much higher spatial sampling rate was desired for this WSN so that high-quality data could be collected from a small volume of space. These conditions are akin to those in industrial settings, e.g. factories. To increase flexibility, changing the type of sensor should be a relatively easy task.
- *Environment features:* The dimensions of the environment in which the nodes operate should be large enough to allow complex experiments, but not so large that access to the nodes is problematic. Real-time control over the environment using sensory data is desirable. It should also be possible to modify it beyond the original specifications to facilitate new kinds of initially unforseen experiments.

- *Communication characteristics:* Losing packets is unavoidable when using real-time constrained transmissions. Additionally, wired connections are usually more reliable than wireless communication links, but they need extra infrastructure.
- *Cost:* The test-bed should be optimised to give the best ratio of performance to price. This should include measures such as increasing the work a single node is capable of, using as few extra components (especially costly ones) as possible, and allowing for reuse of existing resources.

3 Desired Features and Their Realisation

There are four main elements in a wireless sensor network test-bed: the sensing hardware, the control software, the support software, and the environment in which the experiments will be performed. There are many choices for sensor node hardware; there exist a range of devices from small 8-bit, memory-poor units (e.g. motes [1]) to 32-bit, WiFi enabled microservers (e.g. Stargates[3]). These devices are often modular, facilitating the connection of different radios or sensor arrays, depending on the task at hand. Control software, that is the software the sensor nodes run, is usually the primary subject of experiment in WSN test-beds, though there is commonly a permanent administration layer which allows for node control, reprogramming, debugging, etc. Support software, which typically runs on servers that monitor the network, should record the state of the network during experiments, for instance which nodes are active, what messages they are transmitting, what data their sensors are generating, and store it for future analysis. Remote access to the network, job scheduling, and visualisation tools are services that are frequently provided by this element of the test-bed. Finally, the environment in which the experiments takes place is critically important to the test-bed. It can be characterised by being open or closed, static or dynamic, and whether it is controllable by the system or not. The following sections deal with the specifics of each of these aspects of Microcosm (see figure 1).

3.1 Sensing Hardware

There are a number of features that are desirable in a sensor network test-bed, not least of which is adequately dense sensor deployment. Many test-beds use hardware that is confined to having a single sensor per modality per node, i.e. they have a 1-to-1 mapping between sensor types and processing/radio platforms. This ratio can be increased so that one sensor node supports multiple sensors of a single type that are spatially separated. This has the effect of producing higher quality data without increasing costs proportionally. Additionally, flexibility in the placement of the sensors provides the facility to investigate the effects of different deployments on the operation of the WSN.

Given the density of the sensors in this set up, a valid question is why use sensor nodes rather than wiring sensors onto a bus attached directly to a server. The reason why they were employed is because a protocol for e.g. determining necessary sensory coverage needs to be distributed, and preferably local, if it is to work in a WSN. Using a centralised system does not fit with this and would not allow for these kinds of protocols to be tested, so we did not use this method in Microcosm.

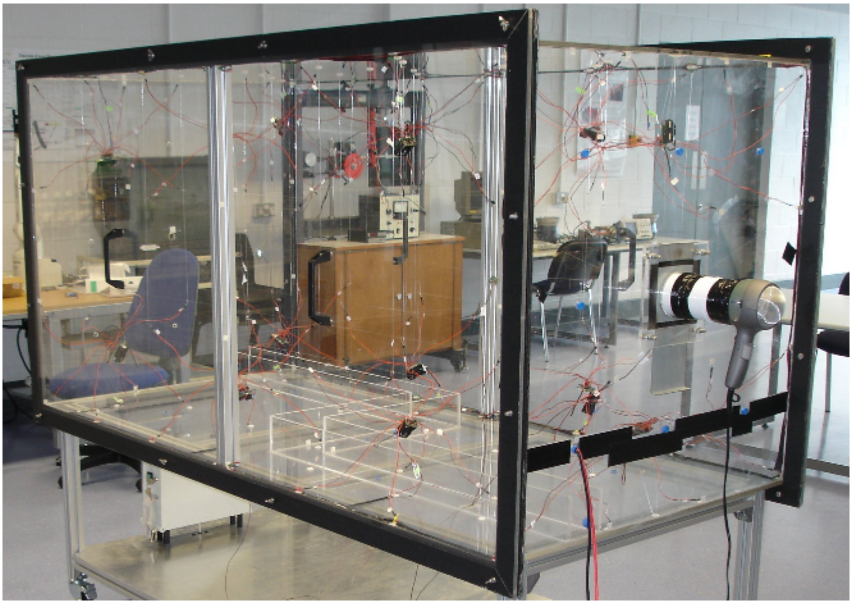

Fig. 1. Microcosm includes a test chamber and a set of sensor nodes deployed within it

Secondly, a test-bed should be able to run experiments that require different kinds of sensors. The ability to easily change the type of sensor that is being used allows for a much greater range of experiments to be carried out. Typically this is achieved by swapping one sensor board for another, however oftentimes much of the hardware is duplicated across these different boards, and so a method to reduce this redundancy would save on costs. What is needed is a plug-and-play sensing capability.

Implementation. Hardware meeting these requirements was devised to work with Mica 2 motes [1] running TinyOS [4]. Briefly, these devices have 7.38 MHz, 8-bit processors, 4 kbytes of RAM, 128 kbytes of program memory, 512 kbytes of flash memory and 19.2 kbit/sec radios. They have 8 ADC channels with 10-bit resolution and an expansion connector to which sensor boards can be attached. While prefabricated sensor boards are available, for the purposes of this set up, they do not satisfy the above constraints.

These boards used a number of features to meet the above requirements. Each one has eight individual sensors, with the sensors divided into four pairs, each pair being wired to a single ADC channel. This was necessary since one of the eight ADC channels is already connected to the radio to measure received signal strength intensity, and so only seven remain available. By ensuring power is only supplied to one sensor in each pair at a time (with an appropriate delay between switching between sensors and sampling the ADC to avoid mixed signals), sensors can share channels without interference. A single board could potentially have far more sensors, however this would require multiplexors, which would increase the complexity and cost of the hardware.

The sensors themselves lie at the end of lengths of wire approximately 30cms long that allow the sensors to be placed at various distances from the node itself. The wires include plug-like terminations (figure 2) to facilitate the replacement of one sensor type with another, for example thermistors (the sensor used in the current incarnation of Microcosm) with light dependent resistors. This fulfills the need for easy alteration of the sensing modality of the test-bed without unnecessary duplication of resources. With eight sensors, it is possible to put each at the corner of a cube around the sensor

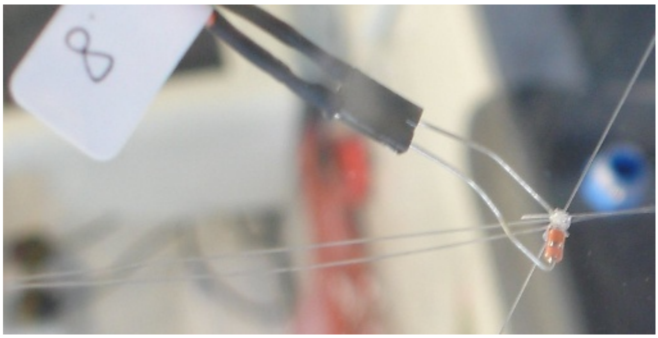

Fig. 2. A thermistor plugged into one of the multi-purpose sensor receptacles on the Microcosm's custom sensor boards. It is held in place by 0.2mm nylon lines.

platform, allowing for a much greater spatial coverage than if the sensors were attached to the sensor board itself. With this arrangement, it is possible for even a single node to gain directional data about a stimulus. Thermistors were used as sensors in this set up, and even with a weak heat source such as a light bulb placed roughly 20cms from the nearest sensor, a 3 K difference was registered between thermistors close to the bulb versus those that were more distantly located. This difference is large enough to be useful given that the combination of these thermistors and the Mica2's 10-bit ADC has a sensitivity of better than 0.1 Kelvin at room temperature.

Problems and Solutions. A number of problems arose during the design of these components.

– *Cost of low run components:* Creating a small number of circuit boards often incurs a high cost. However, when the number is small, it is possible to manually construct the boards. There are a number of methods which could be employed, however the experience gained from the fabrication of the boards for Microcosm indicates that the best method is to use commercially available copper-board etching kits. A single template can be used to produce multiple boards relatively quickly. The only caveat is that, because of the connectors used on the Mica2, the feature size is small enough to require a high level of manual precision during fabrication.

- *Calibration:* The thermistors used in this design had a tolerance of 5%, which translated to roughly ± 1 Kelvin. Calibration was necessary to transform the raw readings into usable data. Since the sensors were deployed in a controlled environment, it was possible to measure the temperature of the interior of the chamber using thermocouples, and using these reference values, adjust the manufacturer-provided resistance-to-temperature equation (1) to correct for the deviation of each sensor.

$$T_{kelvin} = \frac{1}{a + b * ln(R) + c * ln(R)^2 + d * ln(R)^3} \tag{1}$$

  where R = resistance of thermistor, $a = 3.359908 * 10^{-3}$, $b = 2.5788772 * 10^{-4}$, $c = 2.5364809 * 10^{-6}$ and $d = 5.3216393 * 10^{-8}$
- *Unreliable radio communications:* The radio is an unreliable channel. This means data will be lost if sent using it alone. Other test-beds have employed wired channels to improve reliability with positive results (see section 4), and so the same method is being adopted for this WSN. At this point, a small-scale prototype has been built to evaluate the design concepts. The wiring up of the test-bed itself is part of the imminent upgrades for Microcosm (see section 5).
- *Part availability:* Though not encountered during the construction of Microcosm, an obstacle to any future attempt to build a similar system based on hardware that is not very new is the poor availability (for instance, due to RoHS non-compliance) of some components. Any test-bed projects which intend on creating custom parts are strongly advised to choose hardware which will not suffer from this problem.

3.2 Sensor Node Control Software

There are some features of sensor node control software that most WSN test-beds have in common. First is the ability to report back data about their operation. In the context of a test-bed tailored towards sensory experiments, transmitting sensor readings is critical. Having real-time access to this data allows the support software (discussed below) to alter the environmental conditions as part of a closed control loop. Reprogramming the nodes is another useful feature. This can happen either over the radio or via a direct physical connection. Wireless based schemes can often perform quite slowly compared with those set ups that use a wired infrastructure to reprogram the nodes, however they require significantly less hardware to enable, reducing deployment cost and time.

A mechanism for keeping the nodes synchronised also helps with collecting useful sensory data. Even if they start off synchronised, wireless sensor nodes often experience moderate clock skew, and so the nodes can not be relied upon to maintain synchronisation for any extended period. Again, there exist wireless methods of keeping the nodes in step, but it is preferable that the overhead is small so that the wireless channel can be focused on transmitting data of interest. Related to this is the necessity of ensuring maximal efficiency when transmitting data. It is generally accepted that TDMA protocols are effective at increasing efficiency, and that multihop networking protocols reduce overall throughput to the benefit of per node energy consumption.

Implementation. For experiments that are primarily concerned with sensing, the software to control the motes does not need to be particularly complex. Because networking was not the primary concern in the design of Microcosm, currently a simple TDMA MAC layer is used. No multihop network protocol was used because (a) the test chamber is relatively small, (b) overall packet throughput is increased and (c) it would unnecessarily increase the complexity of the software. A few simple commands allow for moderate flexibility in the operation of the network, while employing wireless reprogramming capabilities gives the option of more extensive, though slower, changes to the control software. By identifying the commonly changed parameters of the software, it is possible to construct command packets to change these parameters, thus effecting large changes with a minimum of time and effort. The properties that were found to be most useful to change were:

- The sampling rate of the sensors
- The number of attempts to resend a packet over the radio that should be attempted (for increased reliability)
- The delay between nodes transmitting their packets, and
- The power at which the radio transmits

Since Mica2 motes are subject to clock skew, they cannot be relied upon to remain synchronised for any length of time without a correcting mechanism. The solution used in this test-bed was to sample the sensors when a trigger packet is received from the base station. To reduce overhead, the above mentioned commands were incorporated into this packet. The timing of the transmission of this packet is governed by software running on a desktop computer, and so has far more accuracy than the motes' clocks. Additionally, this allows the rate at which the sensors are sampled to be changed easily.

Once the trigger message is received from the base station, each node samples its sensors. It then waits for a time interval equal to the product of its ID number and the period specified in the trigger packet. This gives each node a unique time slot in which to transmit. By altering the length of this period, the throughput of the network can be configured, either to speed reception of the packets, or to allow more time so that repeat packets can be sent before the next node is due to transmit (as a reliability mechanism). In other words, the TDMA protocol can be optimised for different tasks depending on the particular requirements of the application or experiment. However, the real-time aspect of the system is preserved regardless of where the emphasis is placed.

Because there are multiple sensors connected to one node, it makes sense to collect readings from all sensors and transmit them in a single packet. This cuts down on the number of packets sent through the radio channel, as the overhead associated with the packet headers is now divided over 8 individual readings instead of just one. While a single-sensor node could buffer readings until it had enough to fill a standard 29 byte payload, this would impinge on the real-time aspect of Microcosm. Real-time operation is also the reason why data is not stored locally on nodes to be transmitted later. Transmitting all eight readings in a single packet produces an effective increase in the data extraction rate of the network, and allows the sensors to sample at a higher rate, thus improving the quality of the data collected.

Lastly, the strength at which the radio transmits can be altered. This can be useful when trying to eliminate packet losses. Losses can occur if the signal strength is too low, and the packet fails to register with the receiving mote, or when it is too high, and reflections of the signal interfere with the reception of the packet.

Problems and Solutions. The unreliability of the wireless channel was the principle cause of problems for the sensor node control software.

- *Packet Loss:* One major issue regarding the collection of data from the network is the loss of packets that is unavoidable with wireless links. While mechanisms like acknowledgements, negative acknowledgements, etc. can be used to try to increase the likelihood that a packet will be received, it is impossible to guarantee reception. Because the nodes were expected to send data regularly, there was not much time for attempted retransmissions. It was also found that packets were most often missed because of a physical obstacle, even a person, causing an adverse effect on the signal path. This often meant that no amount of retransmission would work until the physical cause was removed. One solution to this problem is to store all sensor data in the flash memory on the nodes. Any gaps in the data received could be requested once the experiment has finished. Without the time constraints associated with real-time data streams, the node could continuously retransmit until the packet eventually gets through.
- *Wireless Reprogramming:* This can be a slow process. It was found that when using Deluge, the TinyOS network reprogramming tool, even a fairly small program could take many hours to distribute over the network. While this is not a problem if the reprogramming can be set to run overnight, there are times when more immediate action is necessary. In a deployment of a small number of motes, it can be more effective to reprogram each of them directly with a standard programming unit.

3.3 Support Software

Generating copious data is pointless without a sensible means of collecting and reusing it. Different algorithms should be evaluated using the same set of data, which is impossible to guarantee across separate runs of a physical experiment. Logging data generated when the sensor network is exposed to an environmental stimulus, then running a battery of test algorithms on the stored data, is one way of ensuring that there are no discrepancies between runs when performing the comparison. Testing multiple algorithms on live data would potentially require the time consuming process of running multiple instances of the same experiment serially. Instead of this approach, by testing algorithms on recorded data, the requirement of rerunning an experiment is removed, and the different algorithms can potentially now be tested in parallel.

In conjunction with logging sensed data, it is also of paramount importance that the data be monitored as it is logged to ensure its integrity. For example, it would be unwise to run and log a lengthy experiment if a number of the nodes are not capable of transmitting to the base station either due to obstacles, incorrect aerial orientation or other environmental factors. The support software should also provide a mechanism for examining the fidelity of this data while the experiment is running. This can potentially allow an administrator of an experiment to tweak the sensor nodes' control software (discussed earlier) to deliver optimal performance. An example of this control could be increased transmission frequency or alternate routing of packets to avoid obstacles.

Implementation. To deliver this functionality, a number of support tools were implemented for use on one or more of the base stations of Microcosm. When implementing the *logger*, the requirement of data replay requires the time-stamping of event messages received at the base station. Upon receipt of a radio message, the time of arrival of the message and the raw message itself is logged to a file. The packet received is also displayed to the user, this can give a rudimentary indication of faulty nodes or nodes unable to transmit to the base station. When the experiment is completed, the logger is stopped and the file of logged data is submitted through a server script to a web server. This functionality allows sharing of data between multiple team members transparently and without conscious effort on the part of the user. The logs are then available through a web page for download. Another feature of this is the ability to notify one or more team members when a log is uploaded through experiment completion alters, delivered via email. This allows users interested in other team members' experiments to have access to the logs as soon as they become available.

Once the logging process is completed and the logs have been archived, another tool, the *player*, can be used to process the logged files and generate the original packets at the appropriate time intervals. The player parses the file to calculate the time difference between logged packets and can replay the information in one of three modes:

1. Play: it can maintain the absolute time difference between events
2. Accelerated play: it can pre-process the logged file to discover if it is possible to maintain the relative time duration between played events, or
3. Fast forward: it can play the events as fast as possible.

Each of these approaches is arranged in order of increasing speed of execution and reducing temporal fidelity with respect to the original experiment. Some experiments may not be sensitive to the temporal aspect of an experiment and may instead be concerned solely with the contents of the logged packet.

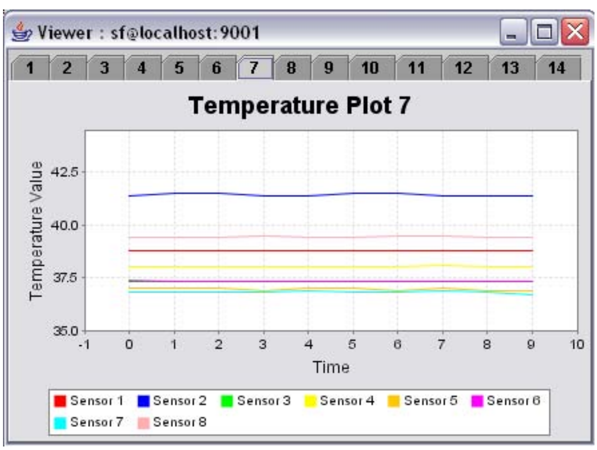

Fig. 3. A screen shot of the viewer tool

The final component of the support software toolset is the *viewer*, use to display the messages in real time as they arrive at the base station. Figure 3 shows an instance where the viewer is displaying the temperature readings of the thermistors appended to the individual nodes. There are 14 nodes in the test-bed, leading to 14 tabs in the viewer which correspond to each node's id. Eight trends are displayed on the graph and these allow the user to visually analyse the data coming from individual nodes. If there is a flaw in a sensor, this can be detected in the trends and can point the user to the node that needs attention.

The support software is a crucial part of a WSN test-bed and it primarily consists of two parts. The first part, such as the logger and player, is relatively general and can be reused for many different applications. On the other hand there are application specific components, such as the viewer, which interpret the data for the user and give an indication of the specific performance of this particular experiment. Most practical test-beds will require both types of support software.

3.4 Experimental Environment

A test chamber within which the operation of various sensors and sensor networks could be analysed has been designed and constructed. A number of critical design requirements were adhered to. In order to allow for video capture equipment and to give multiple viewing angles for demonstration purposes, the unit is constructed from bonded 10mm thick clear Perspex^{TM} sheets and has a dimension of 2x1x1m. The internal volume of the chamber is therefore 2m^3. The applications for sensor networks are not restricted to a purely gaseous environment. To give the test-bed all-round functionality a channel was incorporated to run along the base of the unit allowing for testing in liquid

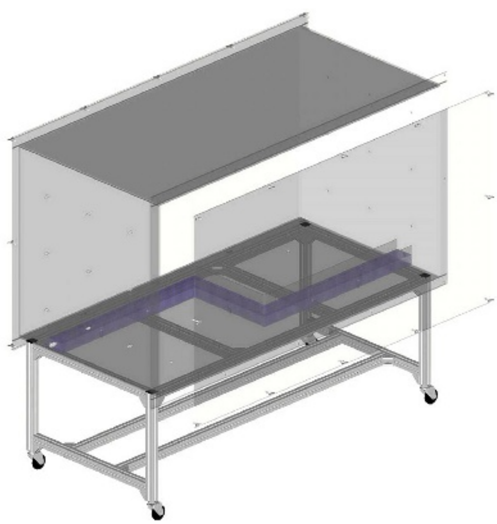

Fig. 4. Rendered view of the test chamber

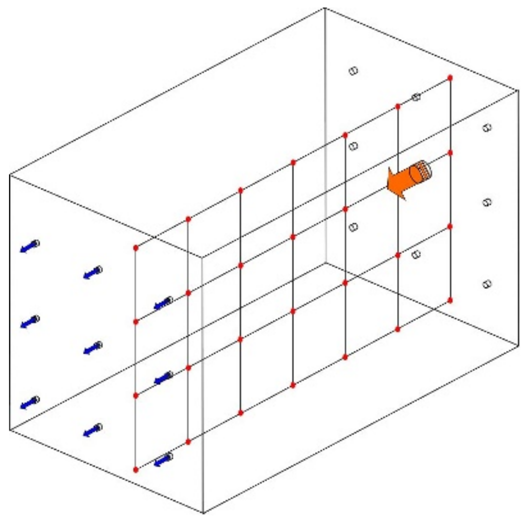

Fig. 5. A cross section through the support lattice for the sensors. Also shown are some of the air vents which allow for gases to be injected into the chamber.

environments. There are multiple inlet/outlet ports to allow seeding of the environment. A rendered representation of the chamber is shown in figure 4.

Implementation. The chamber has been adapted to suit the temperature monitoring requirements of the WSN under discussion. However testing with other sensor platforms and external heat sources was carried out concurrently. An appropriate sensor layout was decided upon to best suit the dimensions of the chamber. The sensors were arranged in a 4 x 4 x 7 matrix (a density of 56 nodes/m^3). One vertical plane of the matrix is shown below in figure 5. There are 14 nodes each consisting of 8 sensors as discussed earlier, resulting in a total of 112 individual sensor points.

Problems and Solutions. The large quantity of sensor devices to be placed within the chamber posed problems, initially due to the accurate positioning of the devices themselves and subsequently due to the reduced accessibility within the chamber with the sensors are in position.

- *Sensor Positioning:* The sensors were located by attaching them to a wire frame that was constructed within the chamber. The frame was made up of a thin gauge (0.2mm) nylon wire (see figure 2) that was tied to anchoring points on the internal faces of the chamber. The anchoring points had been accurately positioned so that the sensors would be evenly spaced with 333mm between each sensor in the X, Y and Z-axes. The anchors used were 10mm x 10mm sticky-back cable tie bases (RS part no. 666-751). Tying together points where two lengths of wire crossed reinforced the lattice. The Mica2 motes positions were also fixed within the chamber using a similar method.

– *Power supply to heat sources:* A control system has been developed which allows for the operation of external heat sources via circuitry operated from the parallel port connection on a laptop PC. The circuit being used to operate a light is shown in figure 6. There are 8 data pins available via the computers parallel port. This circuit layout can thus be replicated up to eight times to control eight individual devices. When the data pin is set high the transistor is activated allowing the 24V DC supply to pass through the coil of relay #1, switching its contacts. 24V is then in turn connected across the coil of the mains relay. The contacts in this relay are closed, connecting the mains power supply to the light. When the data pin is returned to low the transistor is deactivated. No power can therefore flow through both coils, and thus removes the magnetic effect on the switches, thus opening them and breaking the circuit. As this can be run from the base station, data from the sensors can govern the operation of this circuitry, providing a closed control loop.

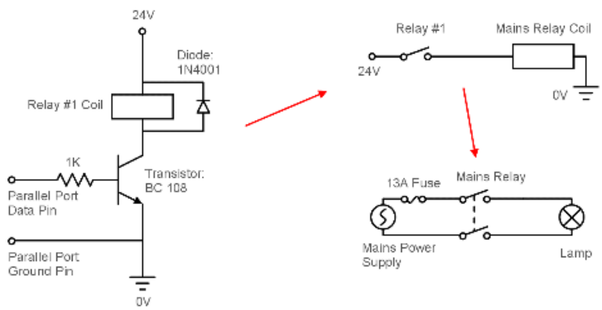

Fig. 6. A diagram illustrating some of the control circuitry used to operate the test-bed remotely

– *Power supply to motes:* For any experiment, it is essential to remove any operational variables which are not under scrutiny. The response of the sensors is dependent on the current state of the battery voltage present in the Mote devices. This problem was overcome by hardwiring a 3V power supply to each of the 14 devices. A parallel supply circuit was employed so that if there were a fault on one of the devices that only that particular device would be removed from the network. This setup also removed the necessity to change discharged batteries that would have proved cumbersome within the constrained space of the chamber. However, batteries may still be employed if it is experimentally necessary.

One of the most important recommendations arising from the experiences in constructing Microcosm is to allow for sufficient flexibility to meet changing requirements during the construction process. An iterative approach was adopted for the design of the separate elements of the test-bed. For example, the mote-based hardware went through several stages, the first incorporating multiple sensors, the second facilitating sensor swap-out, the third adding wired communications ability, and so on. Without modifiable initial designs, much of the early hardware would had to have been replaced in order to allow the later improvements to be made.

4 Related Work

There are several other WSN test-beds described in the literature. Some emphasise the networking aspect, while others have more extensive sensing capabilities. They can be distinguished by a number of features: combined wired and wireless vs. wireless only communications, the presence/absence of feedback-based control, easy reprogrammability, remote access to the test-bed/data, as well as some more unusual abilities such as automated experimentation, capacity for node heterogeneity, mobile elements, hierarchical structuring and 3-D sensor arrangement. All the systems below are situated in open environments and have sensor resolutions far lower than Microcosm.

Motelab [5] is a system offering a set of tools for managing a network of a few dozen motes deployed over 3 floors of the the Electrical Engineering and Computer Science building in Harvard. These tools allow users to create and schedule tasks on the network, record data, reprogram the nodes and access previous results through either a web or a database interface. It is unique in that it includes a power consumption monitor for one of the nodes in the network. Alongside this, it employs a wired infrastructure for node reprogramming and data collection, in order to avoid the unreliability of the wireless channel that has been mentioned previously. One downside to this is that each sensor node requires its own reprogramming board, a significant extra cost. A fairness protocol for time allotment between multiple users has also been implemented.

In contrast to the above static deployment, Mobile Emulab [6] uses a combination of six mobile sensors, in the form of Garcia robots with Mica2 motes attached, and 25 fixed motes in an area of 60m^2 to perform experiments using the Emulab [7] test-bed framework. Emulab provides abstractions to facilitate easy creation and scheduling of experiments, and can log test data and debugging information. Mobile Emulab can track the robots by employing static cameras, mote-based magnetometers and onboard sensors. This multi-sensor modality is unusual among test-beds. The robots can be controlled through the Emulab software, and so represent a method for introducing control of the environment into the experiments in a manner quite different from the method described in this paper. It also contrasts with the majority of set ups, in which the data from the sensors does not influence the environment they are in. Because the robots have motes attached, they can act both as stimulus to the fixed network and data collectors for the system as a whole.

TWIST [8] consists of a tiered network of 90 nodes, with subsets of these nodes connected to supernodes via USB, which in turn are connected to a central server through ethernet. It has the capacity to conduct experiments on heterogenous WSNs that conform to flat, segmented or hierarchical topologies by switching the roles the various components play. The basic sensor node can be any USB-enabled device, such as the Telos mote or the eyesIFX. USB allows communications, power and reprogramming to be integrated into a single connection, reducing costs. These are connected to a Linksys NSLU2, an ethernet connected storage and processing unit, which may function either as diagnostic and management devices or may actively participate in the task of the sensor network, thus introducing a second layer into the WSN. The server acts as a control locus for the entire system. For the cost of additional equipment in the form of the Lynxsys devices, this system gains the benefit of a tiered structure.

SensorScope [9] distinguishes itself from other systems in that it does not use a wired infrastructure. The primary reason for this is to increase the realism of the experiments that are carried out on it. Any test-bed that employs fixed links to retrieve data and management information from the sensor nodes in its network increases the amount of work that the nodes must do, and thus the energy they consume. This is because, on top of the usual messages that the nodes transmit to each other over the radio, they must relay additional messages to the monitoring equipment. SensorScope uses only the wireless channel, with a bare minimum of status information messages, to improve the accuracy of their measurements at the expense of gathering less reliable data. This is similar to the current set up used in Microcosm, and it is an option that will be retained once a wired channel is introduced.

Tutornet [10] is similar to the TWIST, in that it has USB-enabled nodes connected to supernodes, in this instance Stargates [3], but is on a smaller scale. As with TWIST, this is to allow rapid reprogramming while leaving the wireless channel free.

Many other test-beds exist, for instance the sMote, Ω and Trio test-beds at Berkeley [11], and the Re-Mote test-bed at Copenhagen[12], however extensive literature on their design and innovations is not openly available.

5 Future Work

Further automation of Microcosm is the primary improvement planned. Ultimately, it would be useful to be able to reprogram the sensor nodes using a wired connection, however this is not a high priority since radio reprogramming simply takes longer. As was mentioned earlier, there is already a prototype of the wired infrastructure that will be integrated into the existing test-bed. This will bring many benefits. By collecting a complete data set, comparisons with the set produced by the radio can be made, allowing measurements of the impact of packet loss on sensor network performance. Additionally, the complete data gives a better view of what is happening in the environmental chamber, and is more useful when comparing fluid dynamic models of the chamber with real readings, this being another of the uses of the project. Remote access to the network will also be vitally important as the number of users of the test-bed grows. To this end we will make use of the internet to coordinate such access. Fine-grained control of stimuli within the environment, such as location, magnitude and spatial extension, would facilitate assisted or even fully automated experimentation in the future. Naturally, Microcosm will form the basis of many WSN experiments, for instance using interpolation as a coverage calculation mechanism [13] and advanced MAC protocol evaluation [14], however detailed discussion of these experiments is beyond the scope of this paper.

6 Conclusion

This paper has described the trade-offs involved in the construction of a low-cost wireless sensor network test-bed, the focus of which is experiments in the sensing domain and the closing of the sense-process-act loop. A discussion of the required features and the hardware and software designs that implement them was used to illustrate the obstacles that can arise in the construction of a WSN test-bed of this sort. Details of the solutions to various problems that were encountered were related in order to provide future projects looking to construct a test-bed with time- and resource-saving knowledge.

Acknowledgements

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361. We would also like to thank the Irish Research Council for Science, Engineering and Technology (IRCSET) for their support.

References

1. Mica2 motes: (http://www.xbow.com/products/productsdetails.aspx?sid=62)
2. SmartIts Website: (http://www.smart-its.org)
3. Stargate microservers: (http://www.xbow.com/products/productsdetails.aspx?sid=85)
4. Hill, J.: System Architecture for Wireless Sensor Networks. PhD thesis, UC Berkeley (2003)
5. Werner-Allen, G., Swieskowski, P., Welsh, M.: MoteLab: A wireless sensor network testbed. In: In Proceedings 4th Int'l Conf. Information Processing in Sensor Networks (IPSN '05), IEEE (2005) 483–488
6. Johnson, D., Stack, T., Fish, R., Flickinger, D.M., Stoller, L., Ricci, R., Lepreau, J.: Mobile Emulab: A Robotic Wireless and Sensor Network Testbed. In: 25th Conference on Computer Communications (IEEE INFOCOM 2006). (2006)
7. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. In: OSDI02, Boston, MA (2002) 255–270
8. V.Handziski, Köpke, A., Willig, A., A.Wolisz: Twist: A scalable and reconfigurable wireless sensor network testbed for indoor deployments. Technical Report TKN-05-008, Telecommunication Networks Group, Technische Universität Berlin (2005)
9. Schmid, T., Dubois-Ferrire, H., Vetterli, M.: Sensorscope: Experiences with a wireless building monitoring sensor network. In: Workshop on Real-World Wireless Sensor Networks (REALWSN'05). (2005)
10. Tutornet Website: (http://enl.usc.edu/projects/tutornet)
11. Testbeds at the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley: (http://www.millennium.berkeley.edu/sensornets/)
12. Re-Mote Testbed, Department of Computer Science, University of Copenhagen: (http://www.distlab.dk/remote/remote.html)
13. Tynan, R., O'Hare, G.M.P., Marsh, D., O'Kane, D.: Interpolation for wireless sensor network coverage. In: EmNetS-II: The Second IEEE Workshop on Embedded Networked Sensors, IEEE (2005)
14. Ruzzelli, A., Cotan, P., O'Hare, G.M.P., O'Grady, M.J., Tynan, R.: Merlin: A synergistic integration of mac and routing protocol for distributed sensor networks. In: Third Annual IEEE Communication Society Conference on Sensor, Mesh and Ad-Hoc Communications and Networks (SECON 2006). (2006)